Practitioner's Docket No. 10018215-1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Michael S. Allison et al.	Confirmation No: 9960
Serial No.:	09/917,377	Examiner: Khanh B. Pham
Filing Date:	07/28/2001	Group Art Unit: 2166
Title:	METHOD FOR PROCESSING EVENTS FROM ELECTRONIC ARCHITECTURE	

Mailstop: Appeal Brief - Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Introductory Comments

Pursuant to the provisions of 37 C.F.R. § 41.30 *et seq.*, the Appellant hereby appeals to the Board of Patent Appeals and Interferences (hereinafter "the Board") from the claim rejections issued in the final Office action dated January 12, 2007 (hereinafter "the final Office action"). A notice of appeal was filed on April 12, 2007, in conjunction with a pre-appeal brief request for review.

Real Party In Interest

The real party in interest is Hewlett-Packard Company by virtue of an assignment of the present application to Hewlett-Packard Development Company, L.P., which is a wholly-owned subsidiary of Hewlett-Packard Company.

Related Appeals and Interferences

There are no prior or pending related appeals or interferences.

Status of Claims

Claims 1-7 and 9-20 are pending in the application.

Claim 8 was canceled in a previous response.

Claims 1-7 and 9-20 have been finally rejected.

Claims 1-7 and 9-20 are being appealed.

Status of Amendments

No claim amendments have been filed subsequent to the final Office action.

Summary of Claimed Subject Matter

Independent claim 18 provides a system for processing events from electronic architecture, wherein the architecture includes a plurality of entities generating the events. In one example, the electronic architecture may be a complex server or other computer system. (Paragraph [0002].) In that case, the entities of the architecture may include processors, operating system (OS) software, programmable devices, firmware files, input/output (I/O) drivers, and electronic sensors and monitors. (Paragraph [0002].) One or more of these entities may generate "events," such as chassis logs indicating the status or "health" of the entity generating the event. (Paragraph [0003].)

Fig. 2 provides one example of such a system 100 coupled with an electronic architecture 104, such as a server. (Paragraph [0018].) In one embodiment, the system 100 includes a computer having an extraction tool (identified as the "getec processing section" 102 in Fig. 2). The extraction tool 102 is configured to extract events, such as chassis logs from the architecture 104. (Paragraphs [0008], [0018], [0019] and [0022]-[0024]. See also operations 212-226 of Fig. 3A for an example.) The extraction tool 102 also separates the events according to the entities of the architecture. (Paragraphs [0024]-[0026]; and operations 228-258 of Figs. 3A and 3B.) The extractions tool 102 also transforms the events into one or more text strings. (Paragraphs [0026]-[0028]; and

operation 252 of Fig. 3B.)

Also shown in Fig. 2, a plurality of analyzers 120A-120F is coupled to the extraction tool 102. (Paragraph [0027].) Also, an interface is provided which couples the extraction tool 102 to either or both of the architecture 104 and a log file 109 configured to store the events from the architecture 104. (Paragraphs [0018], [0023] and [0024]; and operations 216 and 226 of Fig. 3A) The extraction tool 102 is configured to transmit each of the text strings to one of the plurality of analyzers 120. (Paragraph [0027]; and operation 252 of Fig. 3B.) In turn, each of the analyzers 120 is configured to analyze the text strings received from the extraction tool 102 to produce a human interpretable statement summarizing at least one of the events associated with the text strings. (Paragraph [0030]-[0033]; and flow chart 300 of Figs. 4A and 4B.)

Similarly, independent claim 1 provides a method for processing events from electronic architecture, wherein the architecture has a plurality of entities generating the events, as described above. In the method, the events are extracted from the architecture. (Paragraphs [0008], [0018], [0019] and [0022]-[0024]; and operations 212-226 of Fig. 3A.) The events are then separated according to the entities of the architecture. (Paragraphs [0024]-[0026]; and operations 228-258 of Figs. 3A and 3B.) The events are also transformed into one or more text strings. (Paragraphs [0026]-[0028]; and operation 252 of Fig. 3B.) The text strings are then analyzed to produce a human interpretable statement summarizing at least one of the events associated with the text strings. (Paragraph [0030]-[0033]; and flow chart 300 of Figs. 4A and 4B.)

Dependent claim 13, which depends from independent method claim 1, provides for controlling one or more of the extracting, separating, and transforming operations via one or more command line options by updating the command line options automatically from the architecture. (Paragraph [0020].) In one example, this functionality may be provided by way of a command line option indicating the name of a configuration file to be used. (See the "config" command line option of Table 1, pages 4 and 5.) In turn, the configuration file specifies a location, such as an Internet Protocol (IP) address of the architecture, from which the various command line options may be retrieved automatically. (See the "location" configuration file command of Table 2, pages 6 and

7.)

Various embodiments of the invention as claimed allow an engineer, technician or untrained personnel to review both the text strings related to the events of the architecture, and the human interpretable statements produced from the text strings, to analyze the relative status or health of the entities of the electronic architecture. (Paragraphs [0007]-[0011] and [0034].). Thus, persons with an interest in the status of the electronic architecture need not troubleshoot the architecture by working directly with the events or chassis codes, which typically contain sets of numbers that are difficult to decipher and interpret without intense training, and require time-consuming analysis of the codes.

Grounds of Rejection to Be Reviewed on Appeal

1. Whether Claims 1-4, 6, 7, 9 and 16 are anticipated under 35 U.S.C. § 102(e) by U.S. Patent No. 6,598,179 to Chirashnya et al. (hereinafter "Chirashnya").

2. Whether Claims 5 and 10-15 are unpatentable under 35 U.S.C. § 103(a) over Chirashnya in view of U.S. Patent No. 6,269,398 to Leong et al. (hereinafter "Leong").

3. Whether Claims 17-20 are unpatentable under 35 U.S.C. § 103(a) over Chirashnya in view of U.S. Patent No. 6,754,704 to Prorock (hereinafter "Prorock").

Argument

Outline

I. Rejections of Claims 1-20 Under 35 U.S.C. §§ 102(e) and 103(a)

   a. Claims 1-7 and 19-20 Are Allowable Because Chirashnya Fails to Teach or Suggest Analyzing One or More Text Strings to Produce a Human Interpretable Statement

   b. Dependent Claim 13 is Allowable Because Leong Does Not Teach or Suggest Updating Command Line Options Automatically from an Electronic Architecture

I. Rejections of Claims 1-20 Under 35 U.S.C. §§ 102(e) and 103(a)

Claims 1-4, 6, 7, 9 and 16 stand rejected under 35 U.S.C. §102(e) as being anticipated by Chirashnya. (Page 2 of the final Office action.) Claims 5 and 10-15 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Chirashnya in view of Leong. (Page 5 of the final Office action.) Also, claims 17-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Chirashnya in view of Prorock. (Page 8 of the final Office action.) The Appellant respectfully requests reversal of the rejections in light of the arguments presented below.

a. Claims 1-7 and 9-20 Are Allowable Because Chirashnya Fails to Teach or Suggest Analyzing One or More Text Strings to Produce a Human Interpretable Statement

Generally, Chirashnya discloses an error log analyzer (ELA) which "scans error logs generated by a computer system." (Column 2, lines 6-8.) The ELA processes the error log data in three stages: (1) a selection stage (selecting errors which are "of relevance to fault conditions of interest"); (2) a filtering stage (filtering and combining the errors into events known to be associated with particular fault conditions); and (3) an analysis stage (checking the events to decide whether a fault exists that requires service attention). (Column 2, lines 15-29.)

With respect to independent claims 1 and 18, the final Office action indicates that Chirashnya teaches analyzing the text strings and producing a human interpretable statement summarizing at least one of the events associated with the one or more text strings at column 11, line 57, to column 12, line 20. (Pages 3 and 9 of the final Office action.) More specifically, the final Office action states "that at least the 'result number' is [a] text string. For example, Table IV at Col. 12 shows the result number is text strings 'R1' and 'R2'." (Page 11 of the final Office action.)

The Appellant respectfully disagrees. Instead, Chirashnya discloses a Results Table 42 (shown in Fig. 1), wherein each line of the table holds a *result number*, a corresponding *message number*, and a *Boolean indicator* as to whether a field-replaceable unit (FRU) should be reported. (See Table IV at column 12, lines 5-15.) A

separate message catalog holds the messages to be provided to the user, wherein each message is associated with one or more of the result numbers of the Results Table. (See the "Catalog File Examples" at column 12, lines 17-19.) In other words, *numbers and Boolean values* are used as input to present the messages to the user.

Further, Chirashnya indicates specifically with respect to Table IV that the first column is "the *entry number* (i.e., the corresponding *Result_Num*)..." Chirashnya further discloses pseudocode equating the variable "Result" with "Result_Num." (See "Result = Result_Num" several times within Table III, found at column 10, line 37.) In addition, the specific examples listed beginning at column 10, line 53, all show the Result variable representing an *integer, not a text string*. (See "Result = 1" at column 10, line 54, "Result = 3" at column 10, line 59, "Result = 5" at column 11, line 2, and "Result = 1" at column 11, line 3.) Since Result is an integer, and Result is equated with a Result Number, the Result Number must always be an integer, *not* a text string; otherwise, a data-type conflict would result. Further, at no point within the text does Chirashnya refer to the result number as a text string, but only as a number. The use of the letter "R" before the result numbers 1 and 2 in Table IV merely helps to distinguish the Result Number column from the "Message Number" and "Report FRU" columns. In addition, if a textual "R" were actually used as part of *every* Result Number, the "R" would not help distinguish one result number from another, thus making the use of a text string worthless in that situation. Moreover, the "Result" values above would not be merely integers, such as 1, 3, 5, and so on, but would also include a text-format value (such as ASCII) for the letter "R" (e.g., the hexadecimal number 52H). The only strings actually discussed in Chirashnya are those relating to the user messages apparently employed in the final Office action as the "human interpretable statement" of claim 1. (Page 3 of the Office action, citing column 12, lines 5-20, of Chirashnya.) Thus, Chirashnya does not teach or suggest *text strings* being analyzed to provide a human interpretable statement, as provided for in claims 1 and 18.

Thus, in light of at least the foregoing reasons, the Appellant contends that amended claims 1 and 18 are allowable in view of Chirashnya, and such indication is respectfully requested.

Claims 2-7 and 9-17 depend from independent claim 1, and claims 19 and 20

depend from independent claim 18, thus incorporating the provisions of their independent claims. Thus, the Appellant asserts that claims 2-7, 9-17, 19 and 20 are allowable for at least the reasons presented above in support of claims 1 and 18, and such indication is respectfully requested.

b. Dependent Claim 13 is Allowable Because Leong Does Not Teach or Suggest Updating Command Line Options Automatically from an Electronic Architecture

Further regarding claim 13, which indicates that a step of controlling one or more steps of extracting, separating and transforming via one or more command line options comprises *updating the command line options automatically from the electronic architecture*, the final Office action states that such a provision is taught in Leong at column 13, line 65, to column 14, line 5. (Page 6 of the final Office action.) The Appellant respectfully disagrees. In the cited passage, Leong indicates that one embodiment therein "provides for the ability of the *network administrator* to *create and automate execution* of complex Telnet commands." (Column 13, line 66, to column 14, line 1; emphasis supplied.) Thus, Leong indicates that the network administrator actually creates the commands which are later executed in some automatic fashion. Such capability does not teach or suggest *updating command line options* at all, much less updating these options automatically from the architecture. As a result, the Appellant contends that claim 13 is allowable for at least this additional reason, and such indication is respectfully requested.

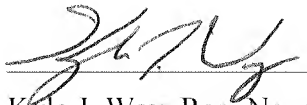
Conclusion

In light of the foregoing remarks, the Appellant submits that the final rejections of claims 1-20 are erroneous, and respectfully requests their reversal.

The Office is hereby authorized to charge Deposit Account No. 08-2025 the requisite fee for this appeal brief (37 C.F.R. §§ 41.37(a)(2) and 41.20(b)(2)). The attendant notice of appeal and fee (37 C.F.R. §§ 41.61(a)(1) and 41.20(b)(1)) were filed previously in conjunction with a pre-appeal brief request for review filed April 12, 2007. Also, the Office is hereby authorized to charge Deposit Account No. 08-2025 the appropriate fee for a one-month extension of time (37 C.F.R. §§ 1.136(a)(1) and 1.17(a)(1)). The Appellant believes that no additional fees are due with respect to this filing. However, should the Office determine that additional fees are necessary, the Office is hereby authorized to charge Deposit Account No. 08-2025 accordingly.

Respectfully submitted,

Date: 7/5/2007



Kyle J. Way, Reg. No. 45,549
Setter Roche LLP
Telephone: (720) 562-2280
E-mail: kyle@setterroche.com

Correspondence address:

CUSTOMER NO. 022879

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

Claims Appendix

The following is a list of claims involved in this appeal:

1. (Previously Presented) A method for processing events from electronic architecture, the architecture having a plurality of entities generating the events, comprising the steps of:

extracting the events from the architecture;

separating the events according to the entities;

transforming the events to one or more text strings; and

analyzing the one or more text strings to produce a human interpretable statement summarizing at least one of the events associated with the one or more text strings.

2. (Previously Presented) A method of claim 1, further comprising the step of filtering the events to process only events from identified entities.

3. (Previously Presented) A method of claim 1, wherein the step of extracting the events comprises extracting chassis logs, wherein the step of separating the events comprises separating the chassis logs, wherein the step of transforming events comprises transforming the chassis logs to text strings, and wherein the chassis logs include chassis codes formed of two numbers.

4. (Original) A method of claim 1, further comprising the step of coupling a getcc extraction tool to the architecture.

5. (Original) A method of claim 4, wherein the step of coupling comprises utilizing telnet.

6. (Original) A method of claim 1, the architecture being a server, and wherein the step of extracting events from the architecture comprises extracting events from the server.

7. (Original) A method of claim 1, wherein the step of transforming comprises converting a binary representation of the events to the text strings.

8. (Canceled)

9. (Original) A method of claim 1, wherein the entities comprises one or more of firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers.

10. (Original) A method of claim 1, further comprising the step of controlling one or more steps of extracting, separating and transforming via one or more command line options.

11. (Original) A method of claim 10, further comprising controlling one or more steps of extracting, separating and transforming according to one or more configuration files.

12. (Original) A method of claim 10, wherein the step of controlling comprises inputting the command line options via a graphical user interface.

13. (Original) A method of claim 10, wherein the step of controlling comprises updating the command line options automatically from the architecture.

14. (Previously Presented) A method of claim 1, further comprising specifying, as command line options, one or more cells of the architecture, and extracting the events only from the one or more cells.

15. (Previously Presented) A method of claim 1, further comprising specifying, as command line options, one or more processors of the architecture, and extracting the events only from the one or more processors.

16. (Original) A method of claim 1, further comprising the step of saving a log file representative of the events.

17. (Previously Presented) A method of claim 1, further comprising the steps of transmitting the text strings to a plurality of analyzers, wherein each of the plurality of analyzers is associated with one or more of the entities, and analyzing the text strings at the plurality of analyzers.

18. (Previously Presented) A system for processing events from electronic architecture, the architecture having a plurality of entities generating the events, the system comprising:

a computer including an extraction tool for extracting the events from the architecture, separating the events according to the entities, and transforming the events to one or more text strings;

a plurality of analyzers coupled to the extraction tool; and

an interface for coupling the extraction tool to one or more of the architecture and a log file storing the events from the architecture;

wherein the extraction tool is configured to transmit each of the one or more text strings to one of the plurality of analyzers; and

wherein each of the plurality of analyzers is configured to analyze the one or more text strings received from the extraction tool to produce a human interpretable statement summarizing at least one of the events associated with the one or more text strings.

19. (Original) A system of claim 18, wherein the entities comprise one or more of firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers, and wherein the events comprise chassis logs from one or more of the firmware, software, processors, architecture monitors, power monitors, cabinet monitors, and I/O drivers.

20. (Previously Presented) A system of claim 18, wherein each of the plurality of analyzers is associated with one or more of the entities.

Evidence Appendix

None

Related Proceedings Appendix

None